Uncertainty Analysis via Failure Domain Characterization: Polynomial Requirement Functions

Luis G. Crespo
National Institute of Aerospace, VA, USA

César A. Muñoz, Anthony J. Narkawicz, Sean P. Kenny, and Daniel P. Giesy
NASA Langley Research Center, Hampton, VA, USA

ABSTRACT: This paper proposes an uncertainty analysis framework based on the characterization of the uncertain parameter space. This characterization enables the identification of worst-case uncertainty combinations and the approximation of the failure and safe domains with a high level of accuracy. Because these approximations are comprised of subsets of readily computable probability, they enable the calculation of arbitrarily tight upper and lower bounds to the failure probability. A Bernstein expansion approach is used to size hyper-rectangular subsets while a sum of squares programming approach is used to size quasi-ellipsoidal subsets. These methods are applicable to requirement functions whose functional dependency on the uncertainty is a known polynomial. Some of the most prominent features of the methodology are the substantial desensitization of the calculations from the uncertainty model assumed (i.e., the probability distribution describing the uncertainty) as well as the accommodation for changes in such a model with a practically insignificant amount of computational effort.

1 INTRODUCTION

This paper studies the reliability of a system for which a parametric mathematical model is available. The acceptability of the system depends upon its ability to satisfy several design requirements. These requirements, which are represented by a set of inequality constraints on selected output metrics, depend on the uncertain parameter vector $\boldsymbol{p}$. The system is deemed acceptable if all inequalities are satisfied. The requirements/constraints partition the uncertain parameter space into two sets, the failure domain, where at least one of them is violated, and the safe domain, where all of them are satisfied. The reliability analysis of this system consists of assessing its ability to satisfy the requirements when the uncertain parameter $\boldsymbol{p}$ is free to take on any value from a prescribed set. The most common practice in reliability analysis is to assume a probabilistic *Uncertainty Model* of $\boldsymbol{p}$ (i.e., the random variable that models the uncertainty), and estimate the corresponding probability of failure. Calculating the failure probability is usually difficult since it requires evaluating a multi-dimensional integral over a complex integration domain. Sampling-based approaches (Niederreiter 1992, Kall and Wallace 1994) and methods based on asymptotic approximations of the failure domain (Rackwitz 2001) are the engines of most (if not all) of the numerical tools used to estimate this probability.

This paper proposes techniques that characterize the uncertain parameter space with a high level of fidelity. A significant thrust of this research is the generation of sequences of inner approximations to the safe and failure domains by subsets of readily computable probability. These sequences are chosen such that they almost surely fill up the region of interest. The strategies proposed, which are only applicable to requirement functions having an explicitly known polynomial dependency on the uncertainty, yield results whose correctness is formally verifiable. The companion paper (Crespo et al. 2011) proposes strategies with the same goal but applicable to unrestricted requirement functions. Overall, the methodology enables the substantial desensitization of the calculations from the assumed uncertainty model as well as the accommodation for changes in such a model with a practically insignificant amount of computational effort.

This paper is organized as follows. Basic concepts are established in Section 2. Section 3 presents strategies for generating and refining approximations to the failure and safe domains. These approximations, along with the developments in (Crespo, Kenny, and Giesy 2011), enable the calculation of upper and lower bounds to the failure probability. Finally, a few concluding remarks close the paper. Proofs are omitted due space limitations.

2 BASIC CONCEPTS AND NOTIONS

Uncertainty models of $\boldsymbol{p} \in \mathbb{R}^s$ can be probabilistic or non-probabilistic. A set whose members are all possible uncertain parameter realizations is a non-probabilistic model. This set, called the *Support Set*, will be denoted as $\Delta \subseteq \mathbb{R}^s$. In a probabilistic uncertainty model, $\boldsymbol{p}$ is a random vector. This model is fully prescribed by the joint probability density function $f_{\boldsymbol{p}}(\boldsymbol{p}) : \Delta \to \mathbb{R}$ or the cumulative distribution function $F_{\boldsymbol{p}}(\boldsymbol{p}) : \Delta \to [0, 1]$.

Consider a system that depends on the uncertain parameter $\boldsymbol{p}$. The design requirements imposed upon such a system are given by the vector[1] inequality $\boldsymbol{g}(\boldsymbol{p}) < \boldsymbol{0}$, where $\boldsymbol{g} : \mathcal{D} \to \mathbb{R}^v$, and $\Delta \subseteq \mathcal{D} \subseteq \mathbb{R}^s$. The set $\mathcal{D}$, where the constraint functions are defined, will be called the *master domain*.

The *failure domain*, denoted as $\mathcal{F} \subset \mathbb{R}^s$, is comprised of the parameter realizations that fail to satisfy at least one of the requirements. Specifically, the failure domain is given by

$$\mathcal{F} = \bigcup_{i=1}^{v} \{\boldsymbol{p} : \boldsymbol{g}_i(\boldsymbol{p}) \geq 0\}. \tag{1}$$

The *safe domain*, given by $\mathcal{S} = C(\mathcal{F})$, where $C(\cdot)$ denotes the *complement* set operator given by $C(\mathcal{X}) = \mathcal{D} \setminus \mathcal{X}$, consists of the parameter realizations satisfying all the design requirements. The failure probability associated with a probabilistic uncertainty model is given by

$$P[\mathcal{F}] = \int_{\mathcal{F}} f_{\boldsymbol{p}}(\boldsymbol{p}) d\boldsymbol{p}, \tag{2}$$

where $P[\cdot]$ is the probability operator. Techniques for approximating $\mathcal{F}$ and $\mathcal{S}$ will be presented below. The resulting approximations are comprised of hyper-rectangles or quasi-ellipsoids.

The *hyper-rectangle* having $\boldsymbol{m} > \boldsymbol{0}$ as the vector of half-lengths of the sides and $\bar{\boldsymbol{p}}$ as its geometric center, is given by

$$\mathcal{R}(\bar{\boldsymbol{p}}, \boldsymbol{m}) = \{\boldsymbol{p} : \bar{\boldsymbol{p}} - \boldsymbol{m} < \boldsymbol{p} < \bar{\boldsymbol{p}} + \boldsymbol{m}\}. \tag{3}$$

An alternative representation of this hyper-rectangle is given by

$$\mathcal{R}(\bar{\boldsymbol{p}}, \boldsymbol{m}) = \delta(\bar{\boldsymbol{p}} - \boldsymbol{m}, \bar{\boldsymbol{p}} + \boldsymbol{m}), \tag{4}$$

where

$$\delta(\boldsymbol{x}, \boldsymbol{y}) = [\boldsymbol{x}_1, \boldsymbol{y}_1] \times [\boldsymbol{x}_2, \boldsymbol{y}_2] \times \cdots \times [\boldsymbol{x}_s, \boldsymbol{y}_s], \tag{5}$$

is the Cartesian product of intervals.

A *subdivision* is the process of dividing a set into subsets. Let $\rho(\cdot)$ be an operator whose input is any given set and its output are the subsets. A bisection-based subdivision in the ith direction is given by

$$\rho(\mathcal{R}) = \{\mathcal{R}(\bar{\boldsymbol{p}} + \boldsymbol{w}, \boldsymbol{m} - \boldsymbol{w}), \mathcal{R}(\bar{\boldsymbol{p}} - \boldsymbol{w}, \boldsymbol{m} - \boldsymbol{w})\},$$

where $\boldsymbol{w} = [0, \ldots, 0, \boldsymbol{m}_i/2, 0, \ldots, 0]$. Alternatively,

$$\rho(\mathcal{R}) = \{\delta(\boldsymbol{v}^1, \boldsymbol{v}^1 + \boldsymbol{m}), \ldots, \delta(\boldsymbol{v}^{2^s}, \boldsymbol{v}^{2^s} + \boldsymbol{m})\},$$

where $\boldsymbol{v}^k$ is a vertex of $\delta(\boldsymbol{l}, \boldsymbol{l} + \boldsymbol{m})$, leads to 2^s rectangular subsets each of volume $\prod_{i=1}^{s} \boldsymbol{m}_i$.

The *quasi-ellipsoid* having $\boldsymbol{m} > \boldsymbol{0}$ as the semi-principal axes vector and $\bar{\boldsymbol{p}}$ as its geometric center, is given by

$$\mathcal{E}(\bar{\boldsymbol{p}}, \boldsymbol{m}, n) = \left\{ \boldsymbol{p} : \left(\sum_{i=1}^{s} \left(\frac{\boldsymbol{p}_i - \bar{\boldsymbol{p}}_i}{\boldsymbol{m}_i} \right)^n \right)^{\frac{1}{n}} < 1 \right\} \tag{6}$$

where n is an even natural number. Note that $\mathcal{E}$ is a closed set in $\mathbb{R}^s$ having a polynomial boundary of degree n. Further notice that $\mathcal{E}(\bar{\boldsymbol{p}}, \boldsymbol{m}, n)$ approaches $\mathcal{R}(\bar{\boldsymbol{p}}, \boldsymbol{m})$ asymptotically from the inside as $n \to \infty$.

When applicable, $\bar{\boldsymbol{p}}$ will be called the *nominal parameter point*. The variable σ will be used to determine the containment of $\bar{\boldsymbol{p}}$ in $\mathcal{S}$. Specifically, $\sigma = 1$ if $\boldsymbol{g}(\bar{\boldsymbol{p}}) < \boldsymbol{0}$, otherwise $\sigma = -1$.

The sections that follow provide the means to generate and sequentially refine approximations of the failure and safe domains.

3 REQUIREMENTS WITH KNOWN POLYNOMIAL DEPENDENCIES

The key development in this section is the calculation of inner and outer approximations to the failure domain. These approximations are comprised of a collection of *almost disjoint* hyper-rectangles or quasi-ellipsoids. The almost disjoint condition is required for estimating failure probability bounds (Crespo, Kenny, and Giesy 2011).

[1] Throughout this paper, it is assumed that vector inequalities hold component-wise, super-indices denote a particular vector or set, and sub-indices refer to vector components; e.g., $\boldsymbol{p}_i^j$ is the ith component of the vector $\boldsymbol{p}^j$.

Let $\mathcal{F}^{sub}$ and $\mathcal{S}^{sub}$ denote inner approximations (sub-sets) of the failure and safe domains. Thus, $\mathcal{F}^{sup} \triangleq C(\mathcal{S}^{sub})$ is an outer approximation (superset) of the failure domain. Note that $\emptyset \subseteq \mathcal{F}^{sub} \subseteq \mathcal{F} \subseteq \mathcal{F}^{sup} \subseteq \mathcal{D}$. Further notice that $C(\mathcal{F}^{sub} \cup \mathcal{F}^{sub})$ contains the failure domain boundary $\partial\mathcal{F}$.

The strategies presented below can only be applied when the dependency of $\boldsymbol{g}$ on $\boldsymbol{p}$ assumes a known polynomial form. This dependency may occur naturally or may be the result of approximations. When the evaluation of $\boldsymbol{g}$ is computationally expensive, it is a common practice to use some realizations of the uncertain parameter and of the corresponding value of the constraint functions to build a surrogate model of $\boldsymbol{g}(\boldsymbol{p})$. If this model is chosen to be polynomial, the approaches introduced hereinafter can be deployed. Note however that the approximation error present in the surrogate model not only undermines the rigorous character of the methodology but may also yield results that are invalidated by the actual $\boldsymbol{g}$.

Bernstein expansion-based techniques (Zettler and Garloff 1998) and interval analysis-based techniques (Jaulin et al. 2001) are well suited for hyper-rectangular geometries. Both of these approaches can be used to approximate sets with hyper-rectangles. The approximations resulting from the latter technique however, are considerably more slack than those of the alternative technique for the same number of subsets. Sum of Squares (SOS) programing-based techniques (Packard et al. 2010) on the other hand, are better suited for sets with polynomial boundaries, e.g., ellipsoids.

Formulations that enable evaluating set containment, determining maximal deformations, and approximating the failure and safe domains are considered subsequently. By set containment we imply determining if all the members of a set are contained by another set. A maximal deformation (Crespo et al. 2008, Crespo et al. 2009) is a process that enables quantifying the separation between a point and $\partial\mathcal{F}$ as well as the identification of worst-case uncertainty combinations. By failure or safe domain approximations we imply the generation of sequences of inner and outer approximations to such sets. The sequence $\{\mathcal{F}_1^{sub}, \mathcal{F}_2^{sub}, \ldots\}$ contains inner approximations of the failure domain. The sequence $\{\mathcal{F}_1^{sup}, \mathcal{F}_2^{sup}, \ldots\}$ contains outer approximations of the failure domain. Similarly, $\{\mathcal{S}_1^{sub}, \mathcal{S}_2^{sub}, \ldots\}$ and $\{C(\mathcal{F}_1^{sub}), C(\mathcal{F}_2^{sub}), \ldots\}$ are sequences of inner and outer approximations to the safe domain. All these sequences approach the domain being approximated when their number of terms increases.

4 BERNSTEIN EXPANSION

The Bernstein expansion approach (Zettler and Garloff 1998) enables determining if a polynomial inequality holds or not over a hyper-rectangular domain. This approach requires mapping this domain to the unit hyper cube. Denote by $\boldsymbol{u} = U(\boldsymbol{p})$ an affine transformation that maps the hyper-rectangle $\mathcal{D}$ onto the unit cube $\mathcal{U} = \mathcal{R}(\frac{\mathbf{1}}{\mathbf{2}}, \frac{\mathbf{1}}{\mathbf{2}})$. Then $\boldsymbol{h}(\boldsymbol{u}) = \boldsymbol{g}(U^{-1}(\boldsymbol{u}))$ is a polynomial on $\mathcal{U}$.

For simplicity in the presentation we first consider the case where there is a single constraint function assuming a univariate polynomial form. If the polynomial $\boldsymbol{h}$ is given by

$$\boldsymbol{h}(\boldsymbol{u}) = \sum_{i=0}^{n} a_i \boldsymbol{u}^i, \tag{7}$$

its Bernstein expansion is

$$\boldsymbol{h}(\boldsymbol{u}) = \sum_{i=0}^{n} b_i(\mathcal{D}, \boldsymbol{g}) B_i^n(\boldsymbol{u}), \tag{8}$$

where

$$B_i^n(\boldsymbol{u}) = \binom{n}{i} \boldsymbol{u}^i (1 - \boldsymbol{u})^{n-i}, \tag{9}$$

is the ith Bernstein polynomial of degree n (i.e., an element of the basis) and

$$b_i(\mathcal{D}, \boldsymbol{g}) = \sum_{j=0}^{i} \frac{\binom{i}{j}}{\binom{n}{j}} a_j, \tag{10}$$

is the ith Bernstein coefficient. Some fundamental properties of this basis are $\sum B_i^n(\boldsymbol{u}) = 1$ (normalization), $0 \leq B_i^n(\boldsymbol{u}) \leq 1$ for $0 \leq \boldsymbol{u} \leq 1$ (boundedness), and $B_i^n(\boldsymbol{u}) = B_{n-i}^n(1 - \boldsymbol{u}) > 0$ (symmetry). Simple manipulations lead to the free function evaluation property:

$$\boldsymbol{h}(0) = b_0(\mathcal{D}, \boldsymbol{g}), \tag{11}$$

$$\boldsymbol{h}(1) = b_n(\mathcal{D}, \boldsymbol{g}). \tag{12}$$

The range enclosing property follows directly from the normalization and boundedness properties and is given by

$$\min_{i \leq n} \{b_i(\mathcal{D}, \boldsymbol{g})\} \leq \boldsymbol{g}(\boldsymbol{p}) \leq \max_{i \leq n} \{b_i(\mathcal{D}, \boldsymbol{g})\}, \tag{13}$$

for all $\boldsymbol{p} \in \mathcal{D}$. Note that the Bernstein expansion enables bounding the range of the polynomial by mere algebraic manipulations. In contrast to the nonlinear optimization approach (Crespo et al. 2011) and SOS

approaches, there is no need to solve an optimization problem or even evaluate the polynomial.

The single constraint, multivariate polynomial case is considered next. Define the multi-index $\boldsymbol{i}$ as a vector of non-negative integers $\boldsymbol{i} = [i_1, \ldots, i_s]$. Note that monomials can be represented as $\boldsymbol{u}^{\boldsymbol{i}} = \boldsymbol{u}_1^{i_1}\boldsymbol{u}_2^{i_2}\cdots\boldsymbol{u}_s^{i_s}$. An s-variate polynomial can be represented as

$$\boldsymbol{h}(\boldsymbol{u}) = \sum_{\boldsymbol{i}\leq\boldsymbol{n}} a_{\boldsymbol{i}}\boldsymbol{u}^{\boldsymbol{i}}, \tag{14}$$

where $\boldsymbol{u} \in \mathcal{U}$. Recall that vector inequalities should be interpreted component-wise. The Bernstein expansion of (14) is given by

$$\boldsymbol{h}(\boldsymbol{u}) = \sum_{\boldsymbol{i}\leq\boldsymbol{n}} b_{\boldsymbol{i}}(\mathcal{D},\boldsymbol{g})B_{\boldsymbol{i}}^{\boldsymbol{n}}(\boldsymbol{u}), \tag{15}$$

where

$$B_{\boldsymbol{i}}^{\boldsymbol{n}}(\boldsymbol{u}) = B_{i_1}^{\boldsymbol{n}_1}(\boldsymbol{u}_1)\cdots B_{i_s}^{\boldsymbol{n}_s}(\boldsymbol{u}_s). \tag{16}$$

is the $\boldsymbol{i}$th Bernstein polynomial of degree $\boldsymbol{n}$ and

$$b_{\boldsymbol{i}}(\mathcal{D},\boldsymbol{g}) = \sum_{\boldsymbol{j}\leq\boldsymbol{i}}\prod_{k=1}^{n}\frac{\binom{\boldsymbol{i}_k}{\boldsymbol{j}_k}}{\binom{\boldsymbol{l}_k}{\boldsymbol{j}_k}}a_{\boldsymbol{j}}, \tag{17}$$

for $\boldsymbol{i} \leq \boldsymbol{n}$, is the $\boldsymbol{i}$th Bernstein coefficient. The normalization, boundedness and symmetry properties extend to the multi-variate case. The free function evaluation property becomes

$$\boldsymbol{h}\left([\boldsymbol{i}_1/\boldsymbol{n}_1, \ldots, \boldsymbol{i}_s/\boldsymbol{n}_s]\right) = b_{\boldsymbol{i}}(\mathcal{D},\boldsymbol{g}), \tag{18}$$

where $\boldsymbol{i}$ is an element of $\{0,\boldsymbol{n}_1\}\times\cdots\times\{0,\boldsymbol{n}_s\}$. The range enclosing property becomes

$$\min_{\boldsymbol{i}\leq\boldsymbol{n}}\left\{b_{\boldsymbol{i}}(\mathcal{D},\boldsymbol{g})\right\} \leq \boldsymbol{g}(\boldsymbol{p}) \leq \max_{\boldsymbol{i}\leq\boldsymbol{n}}\left\{b_{\boldsymbol{i}}(\mathcal{D},\boldsymbol{g})\right\}, \tag{19}$$

for all $\boldsymbol{p} \in \mathcal{D}$. Tighter bounds on the range of $\boldsymbol{g}(\boldsymbol{p})$ are obtained by subdividing $\mathcal{D}$ into subsets, calculating the Bernstein expansion for each of them, and selecting the largest and the smallest of all Bernstein coefficients. In particular, if $\rho(\mathcal{D}) = \{\mathcal{R}_1, \ldots, \mathcal{R}_t\}$ is a subdivision of the master domain, then for all $\boldsymbol{p} \in \mathcal{D}$,

$$\min_{j\leq t}\left\{\min_{\boldsymbol{i}\leq\boldsymbol{n}}\left\{b_{\boldsymbol{i}}(\mathcal{R}_j,\boldsymbol{g})\right\}\right\} \leq \boldsymbol{g} \leq \max_{j\leq t}\left\{\max_{\boldsymbol{i}\leq\boldsymbol{n}}\left\{b_{\boldsymbol{i}}(\mathcal{R}_j,\boldsymbol{g})\right\}\right\} \tag{20}$$

These bounds converge to the global minimum and global maximum of $\boldsymbol{g}(\boldsymbol{p})$ when the volume of the subsets in ρ approaches zero. The reason the bounds in (20) can be calculated efficiently is that the Bernstein coefficients on the elements of $\rho(\mathcal{D})$ resulting from applying the bisection-based subdivision of Section 2 can be calculated directly from those on $\mathcal{D}$ (Zettler and Garloff 1998). The sequential application of the bisection-based subdivision scheme leads to bounds that converge to the global minimum and maximum.

4.1 *Set Containment of Hyper-Rectangles*

The following theorem uses the developments of the previous section to determine whether a set $\mathcal{R}(\bar{\boldsymbol{p}},\boldsymbol{m})$ is fully contained in the safe domain $\mathcal{S}$ or failure domain $\mathcal{F}$.

Theorem 1. *Given a subdivision $\rho(\mathcal{D}) = \{\mathcal{R}_1, \ldots, \mathcal{R}_t\}$, the set containment condition $\mathcal{R}(\bar{\boldsymbol{p}},\boldsymbol{m}) \subseteq \mathcal{S}$ holds if*

$$\max_{k\leq v}\left\{\max_{j\leq t}\left\{\max_{\boldsymbol{i}\leq\boldsymbol{n}}\left\{b_{\boldsymbol{i}}(\mathcal{R}_j,\boldsymbol{g}_k)\right\}\right\}\right\} < 0. \tag{21}$$

Furthermore, $\mathcal{R}(\bar{\boldsymbol{p}},\boldsymbol{m}) \not\subseteq \mathcal{S}$ if there exists $k \leq v$, $j \leq t$, and a multi-index $\boldsymbol{i} \in \{0,\boldsymbol{n}_1\}\times\cdots\times\{0,\boldsymbol{n}_s\}$ such that

$$b_{\boldsymbol{i}}(\mathcal{R}_j,\boldsymbol{g}_k) \geq 0. \tag{22}$$

While Formula (21) results from choosing the largest upper bound in (20) over all the constraint functions, Formula (22) results from applying the free function evaluation property.

Theorem 2. *Given a subdivision $\rho(\mathcal{D}) = \{\mathcal{R}_1, \ldots, \mathcal{R}_t\}$, the set containment condition $\mathcal{R}(\bar{\boldsymbol{p}},\boldsymbol{m}) \subseteq \mathcal{F}$ holds if*

$$\max_{k\leq v}\left\{\min_{j\leq t}\left\{\min_{\boldsymbol{i}\leq\boldsymbol{n}}\left\{b_{\boldsymbol{i}}(\mathcal{R}_j,\boldsymbol{g}_k)\right\}\right\}\right\} \geq 0. \tag{23}$$

Furthermore, $\mathcal{R}(\bar{\boldsymbol{p}},\boldsymbol{m}) \not\subseteq \mathcal{F}$ if there exists $j \leq t$, and a multi-index $\boldsymbol{i} \in \{0,\boldsymbol{n}_1\}\times\cdots\times\{0,\boldsymbol{n}_s\}$ such that for all $k \leq v$,

$$b_{\boldsymbol{i}}(\mathcal{R}_j,\boldsymbol{g}_k) < 0. \tag{24}$$

While Formula (23) results from choosing the largest lower bound in (20) over all the constraint functions, Formula (24) results from applying the free function evaluation property to any of them.

The asymmetry between Formulas (21) and (23) results from the definitions of the safe and failure domains. In order to apply these set containment conditions one is required to implement an algorithm for sequentially subdividing the master domain. When checking whether $\mathcal{R}(\bar{\boldsymbol{p}},\boldsymbol{m}) \subseteq \mathcal{S}$ holds, the subdivision algorithm should be stopped when either (21) or (22) is satisfied. When checking whether $\mathcal{R}(\bar{\boldsymbol{p}},\boldsymbol{m}) \subseteq \mathcal{F}$ holds, the subdivision algorithm should be stopped when (23) or (24) is satisfied. The strategies in (Smith 2009) used to compute Bernstein coefficients were adopted. The computational complexity of the resulting subdivision algorithm is nearly linear with the number of monomials. This makes the Bernstein polynomial approach very efficient.

4.2 *Maximal Deformation of Hyper-Rectangles*

In this section we use the Bernstein expansion approach to evaluate the set containment conditions required to perform homothetic deformations. In particular, one can solve for

$$\tilde{\alpha} = \sup\{\alpha : \mathcal{R}(\bar{\boldsymbol{p}}, \alpha \boldsymbol{m}) \subseteq \mathcal{S}\}, \tag{25}$$

by evaluating the set containment condition with Formula (21) after replacing $\mathcal{R}(\bar{\boldsymbol{p}}, \boldsymbol{m})$ with $\mathcal{R}(\bar{\boldsymbol{p}}, \alpha \boldsymbol{m})$. An identical process can be applied to compute

$$\tilde{\alpha} = \sup\{\alpha : \mathcal{R}(\bar{\boldsymbol{p}}, \alpha \boldsymbol{m}) \subseteq \mathcal{F}\}, \tag{26}$$

using Formula (23).

Alternatively, one may search for the maximal deformation using nonlinear optimization (Crespo et al. 2011) and then use the developments in Section 4.1 to verify convergence to the global optimum, e.g., show that $\mathcal{R}(\bar{\boldsymbol{p}}, \tilde{\alpha} \boldsymbol{m}) \subseteq \mathcal{S}$.

4.3 *Failure Domain Approximations*

The algorithm below iteratively generates indexed sets $\Lambda_i, \mathcal{S}_i^{sub}$, and $\mathcal{F}_i^{sub}$ of hyper-rectangles where $\mathcal{S}_i^{sub}$ is an inner approximation to the safe domain, $\mathcal{F}_i^{sub}$ is an inner approximation to the failure domain, and Λ_i is a region whose containment in $\mathcal{F}$ or $\mathcal{S}$ is to be determined. The application of the set containment conditions (21) and (23) enable the expansion of the inner approximations. The algorithm proceeds by selecting a hyper-rectangle $\mathcal{R} \in \Lambda_i$. If $\mathcal{R} \subseteq \mathcal{S}$, the inner approximation to the safe domain is expanded with this element. If $\mathcal{R} \subseteq \mathcal{F}$, the inner approximation to the failure domain is expanded with this element. If none of these conditions are satisfied, the elements of a subdivision of this rectangle are appended to Λ_i. The algorithm terminates when the bounds to the failure probability exceeds a prescribed limit. The algorithmic representation of this procedure is as follows.

Let $\boldsymbol{g}(\boldsymbol{p}) < \boldsymbol{0}$ denote the set of system requirements and $f_{\boldsymbol{p}}(\boldsymbol{p})$, $\boldsymbol{p} \in \Delta \subseteq \mathcal{D}$ be the uncertainty model. Denote by P_{max} a parameter prescribed by the user denoting the largest admissible failure probability associated with the system. Set $i = 1$, $\Lambda_1 = \{\mathcal{D}\}$, $\mathcal{F}_1^{sub} = \emptyset$, and $\mathcal{S}_1^{sub} = \emptyset$.

1. Let $\mathcal{R}$ be a largest element of Λ_i.

2. Perform a Bernstein expansion of all constraint functions in $\mathcal{R}$.

3. If Equation (21) holds let $\Lambda_{i+1} = \Lambda_i \setminus \mathcal{R}$, $\mathcal{S}_{i+1}^{sub} = \mathcal{S}_i^{sub} \cup \mathcal{R}$, and $\mathcal{F}_{i+1}^{sub} = \mathcal{F}_i^{sub}$. If Equation (23) holds let $\Lambda_{i+1} = \Lambda_i \setminus \mathcal{R}$, $\mathcal{F}_{i+1}^{sub} = \mathcal{F}_i^{sub} \cup \mathcal{R}$, $\mathcal{S}_{i+1}^{sub} = \mathcal{S}_i^{sub}$. If neither equation holds, set $\Lambda_{i+1} = (\Lambda_i \setminus \mathcal{R}) \cup \rho(\mathcal{R})$, $\mathcal{S}_{i+1}^{sub} = \mathcal{S}_i^{sub}$, and $\mathcal{F}_{i+1}^{sub} = \mathcal{F}_i^{sub}$.

4. Let $\mathcal{F}_{i+1}^{sup} = C(\mathcal{S}_{i+1}^{sub})$. Evaluate $P[\mathcal{F}_{i+1}^{sub}]$ and $P[\mathcal{F}_{i+1}^{sup}]$ (Crespo et al. 2011).

5. If $P[\mathcal{F}_{i+1}^{sub}] \geq 1 - P_{max}$ declare the system acceptable and stop. If $P[\mathcal{F}_{i+1}^{sup}] \leq P_{max}$ declare the system unacceptable and stop. Otherwise, increase i by one, and go to Step (1).

As the number of iterations increases, $\mathcal{S}_i^{sub}$ and $\mathcal{F}_i^{sub}$ approach the safe and failure domain. $P[\mathcal{F}_i^{sub}]$ and $P[\mathcal{S}_i^{sub}]$ are monotonically increasing functions of i. On the other hand, $P[\Lambda_i]$ and $P[\mathcal{F}^{sup}]$ are monotonically decreasing functions of i. Note that the elements in Λ_i are an approximation of $\partial\mathcal{F}$. The larger the value of i the better the approximation.

Example 1: Consider the constraint functions

$$\boldsymbol{g}_1 = \boldsymbol{p}_1^2\boldsymbol{p}_2^4 + \boldsymbol{p}_1^4\boldsymbol{p}_2^2 - 3\boldsymbol{p}_1^2\boldsymbol{p}_2^2 - \boldsymbol{p}_1\boldsymbol{p}_2 + \frac{\boldsymbol{p}_1^6 + \boldsymbol{p}_2^6}{200} - \frac{7}{100}, \tag{27}$$

$$\boldsymbol{g}_2 = -\boldsymbol{p}_1^2\boldsymbol{p}_2^4 - \boldsymbol{p}_1^4\boldsymbol{p}_2^2 + 3\boldsymbol{p}_1^2\boldsymbol{p}_2^2 + \frac{\boldsymbol{p}_1^5\boldsymbol{p}_2^3}{10} - 0.9, \tag{28}$$

These requirement functions have been chosen so the safe domain is multiply connected. Figure 1 shows $\mathcal{F}_i^{sub}$ and $\mathcal{S}_i^{sub}$ for a fixed value of i. In this case we have used the bisection-based subdivision where the variable being subdivided alternates. The set of boxes for which set containment cannot be established are colored in white. Note that this region is a tight approximation of $\partial\mathcal{F}$. Tight bounds to the failure probability can be readily calculated from these approximations.

5 SOS PROGRAMMING

Even though the theory supporting the developments that follow applies to all sets having polynomial boundaries, we focus on quasi-ellipsoids since they enable the analytical calculation of their probability (Crespo et al. 2011).

5.1 *SOS Determination*

For simplicity in the presentation we first consider the single constraint case. If $\boldsymbol{g}(\boldsymbol{p})$ is a polynomial of degree less than or equal to $2d$ in the variable $\boldsymbol{p} \in \mathbb{R}^s$, its vectorial representation is

$$\boldsymbol{g}(\boldsymbol{p}) = \boldsymbol{c}^\top \boldsymbol{x}(\boldsymbol{p}), \tag{29}$$

where $\boldsymbol{x} : \mathbb{R}^s \to \mathbb{R}^{\dim(\boldsymbol{x})}$ is a vector of monomials in $\boldsymbol{p}$ of degree less than or equal to $2d$, and $\boldsymbol{c}$ is a vector

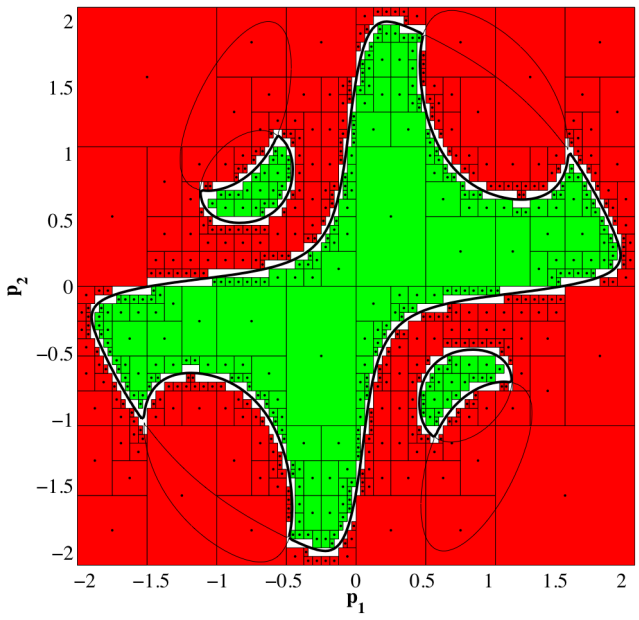

Figure 1: $\mathcal{F}_i^{sub}$ (red), $\mathcal{S}_i^{sub}$ (green), $\partial\mathcal{F}$ (thick line) and individual constraint boundaries (thin lines).

of coefficients. On the other hand, the Gram matrix representation of this polynomial is

$$\boldsymbol{g}(\boldsymbol{p}) = \boldsymbol{z}^\top(\boldsymbol{p}) Q \boldsymbol{z}(\boldsymbol{p}), \tag{30}$$

where $\boldsymbol{z} : \mathbb{R}^s \to \mathbb{R}^{\dim(\boldsymbol{z})}$ is a vector of monomials in $\boldsymbol{p}$ of degree less than or equal to d and $Q \in \mathbb{R}^{\dim(\boldsymbol{z})\times\dim(\boldsymbol{z})}$ is a symmetric matrix. The Gram representation of $\boldsymbol{g}$ is not unique. A procedure to parameterize all possible representations is as follows. Define the linear operator L that maps each symmetric matrix Q to the polynomial coefficients $\boldsymbol{c}$, i.e.,

$$L(Q) = \boldsymbol{c}. \tag{31}$$

A matrix representation of L can be computed since both its domain and its range are finite dimensional. This transformation enables us to parameterize the family of symmetric matrices yielding Gram representations via

$$Q = Q_0 + \sum_{i=1}^{p} \boldsymbol{\lambda}_i N_i, \tag{32}$$

where Q_0 is the symmetric matrix corresponding to a particular Gram representation (i.e., $L(Q_0) = \boldsymbol{c}$), the set $\{N_1, \ldots N_p\}$ is a basis of the null space of L (i.e., $L(N_i) = \boldsymbol{0}$ for $i = 1, \ldots, p$) and $\boldsymbol{\lambda} \in \mathbb{R}^p$ is a vector of multipliers. Note that for any value of $\boldsymbol{\lambda}$ in (32), (30) is a valid Gram representation of $\boldsymbol{g}$.

The Gram representation of a polynomial enables us to determine if a polynomial is a SOS. The polynomial $\boldsymbol{g}(\boldsymbol{p})$ is a SOS if there exist polynomials $h_1, \ldots, h_n$ such that $\boldsymbol{g} = \sum_{i=1}^{n} h_i^2$. The polynomial $\boldsymbol{g}(\boldsymbol{p})$ is a SOS if and only if there exists a positive semi-definite matrix Q, to be denoted as $Q \succeq 0$, that satisfies (30) (Powers and Wormann 1998, Parrillo 2000). The functions $h_1, \ldots, h_n$, that constitute the SOS representation of $\boldsymbol{g}$, result from making Choleski or Schur decompositions of Q. Consequently, $\boldsymbol{g}$ is a SOS if an only if there exist a $\boldsymbol{\lambda}$ for which $Q_0 + \sum_{i=1}^{p} \boldsymbol{\lambda}_i N_i \succeq 0$. This is a Linear Matrix Inequality (LMI) feasibility problem.

Numerical techniques for solving semi-definite programs can be used to find a solution to this LMI feasibility problem. Publicly available software, such as SOSTOOLS, YALMIP and SeDuMi, automate the process of posing and solving this convex optimization problem. Unfortunately, its computational requirements (e.g., the number of monomials that require representation and the dimension of the null space) grow rapidly with the degree of the polynomials and the dimension of the parameter space s.

5.2 *Set Containment of Quasi-Ellipsoids*

We want to determine if the set $\mathcal{E}(\bar{\boldsymbol{p}}, \boldsymbol{m}, n)$ is fully contained in the safe or failure domains by using the developments of the previous section. These set containment conditions, which are equivalent to

$$\mathcal{E}(\bar{\boldsymbol{p}}, \boldsymbol{m}, n) \subseteq \mathcal{S}, \tag{33}$$

$$\mathcal{E}(\bar{\boldsymbol{p}}, \boldsymbol{m}, n) \subseteq \mathcal{F}, \tag{34}$$

can be evaluated using the following Theorem.

Theorem 3. *Let $\mathcal{E} = \{\boldsymbol{p} : e(\boldsymbol{p}) \leq 0\}$, where $e(\boldsymbol{p})$ is polynomial, be an alternate representation of the quasi-ellipsoid in (6), $I = \{1, \ldots, v\}$ and $X = \{i \in I : \boldsymbol{g}_i(\bar{\boldsymbol{p}}) > 0\}$. If there exist a function $\boldsymbol{q}(\boldsymbol{p}) : \mathbb{R}^s \to \mathbb{R}^v$ such that $\boldsymbol{q}(\boldsymbol{p}) \geq \boldsymbol{0}$ and $\boldsymbol{q}_i(\boldsymbol{p})e(\boldsymbol{p}) - \boldsymbol{g}_i(\boldsymbol{p}) \geq 0$ for all $i \in I$ and all $\boldsymbol{p} \in \Delta$ then $\mathcal{E}(\bar{\boldsymbol{p}}, \boldsymbol{m}, n) \subseteq \mathcal{S}$. On the other hand, if there exists a function $q(\boldsymbol{p}) : \mathbb{R}^s \to \mathbb{R}$ such that $q(\boldsymbol{p}) \geq 0$ and $q(\boldsymbol{p})e(\boldsymbol{p}) + \boldsymbol{g}_i(\boldsymbol{p}) \geq 0$ for some $i \in X$ and all $\boldsymbol{p} \in \Delta$ then $\mathcal{E}(\bar{\boldsymbol{p}}, \boldsymbol{m}, n) \subseteq \mathcal{F}$.*

Note that there are 2 positive semi-definite (PSD) constraints per constraint function. PSD constraints, even polynomial ones, are numerically difficult to solve. However, by restricting the constraint functions $\boldsymbol{g}$ to take on a multivariable polynomial form; and relaxing the constraints to be SOS polynomials, in which case Theorem 3 still holds, we obtain a problem that can be solved efficiently. While all SOS polynomials are PSD, not all the PSD polynomials are SOS. Therefore, the relaxation to SOS polynomials yields a sufficient condition for the global nonnegativity constraints in Theorem 3.

In practice, the SOS polynomial multipliers $\boldsymbol{q}(\boldsymbol{p})$ and $q(\boldsymbol{p})$ must be restricted to be in a fixed finite

dimensional subspace of polynomials (e.g., quartic polynomials). This is achieved by prescribing a polynomial basis for them, and using semi-definite programing to search for the coefficients of the corresponding linear combination. Note however that the prescription of an overly restrictive basis may wrongly invalidate the set containment condition (e.g., a quadratic basis for q is prescribed but a quartic q is needed to demonstrate containment). Further notice that the SOS approach cannot be used to demonstrate that there is no set containment. Numerical experiments indicate that problems on the order of $s + n + \sum \deg(\boldsymbol{g}_i) < 12$ are solvable.

5.3 *Maximal Deformation of Quasi-Ellipsoids*

In this section we use the SOS programing approach to evaluate the set containment conditions required by the homothetic deformations. In particular, we want to solve for

$$\tilde{\alpha} = \sup\{\alpha : \mathcal{E}(\bar{\boldsymbol{p}}, \alpha\boldsymbol{m}, n) \subseteq \mathcal{S}\}, \tag{35}$$

$$\tilde{\alpha} = \sup\{\alpha : \mathcal{E}(\bar{\boldsymbol{p}}, \alpha\boldsymbol{m}, n) \subseteq \mathcal{F}\}. \tag{36}$$

Theorem 3 enables the reformulation of (35) as

$$\tilde{\alpha} = \left(\max_{\beta,\, \boldsymbol{q}(\boldsymbol{p})} \beta\right)^{\frac{1}{n}} \text{ subject to} \tag{37}$$

$$\boldsymbol{q}(\boldsymbol{p}) \in \theta[\boldsymbol{p}], \tag{38}$$

$$-\boldsymbol{g}_i(\boldsymbol{p}) + \boldsymbol{q}_i(\boldsymbol{p})\,(e(\boldsymbol{p}) - \beta + 1) \in \theta[\boldsymbol{p}],\ \forall i \in I, \tag{39}$$

$$\boldsymbol{a} - \bar{\boldsymbol{p}} \leq\ ^{1/n}\!\sqrt{\beta}\boldsymbol{m} \leq \boldsymbol{b} - \bar{\boldsymbol{p}}, \tag{40}$$

where $\Delta = \delta(\boldsymbol{a}, \boldsymbol{b})$, and $\theta[\boldsymbol{p}]$ denotes the set of SOS polynomials in the variable $\boldsymbol{p}$. Constraint (40) ensures that the maximal set is fully contained in the support set so its probability can be calculated analytically.

The application of Theorem 3 to (36) yields

$$\tilde{\mu} = \max_{i \in X} \mu_i, \tag{41}$$

where

$$\mu_i = \left(\max_{\beta,\, q(\boldsymbol{p})} \beta\right)^{\frac{1}{n}} \text{ subject to} \tag{42}$$

$$q(\boldsymbol{p}) \in \theta[\boldsymbol{p}], \tag{43}$$

$$\boldsymbol{g}_i(\boldsymbol{p}) + q(\boldsymbol{p})\,[e(\boldsymbol{p}) - \beta + 1] \in \theta[\boldsymbol{p}], \tag{44}$$

and constraint (40). In general $\tilde{\mu} \leq \tilde{\alpha}$. This is so because $\mathcal{E}(\bar{\boldsymbol{p}}, \tilde{\mu}\boldsymbol{m}, n)$ will be limited by one of the manifolds $\boldsymbol{g}_i(\boldsymbol{p}) = 0$ with $i \in X$, but such a manifold may not be part of $\partial\mathcal{F}$. Therefore, while the formulation for containment by the safe domain always converges to the critical similitude ratio $\tilde{\alpha}$, the one for containment by the failure domain may only underestimate it.

The above problems, which have an objective function that is linear in the decision variables (i.e., β) as well as multiple SOS constraints, are called SOS programs. The terms $\boldsymbol{q}_i\beta$ in (39) and $q\beta$ in (44) make both SOS programs bi-linear in the decision variables. The constraints containing these terms are quasi-convex; e.g., for a fixed value of β all constraints are convex in $\boldsymbol{q}$. Hence these problems can be solved sequentially by searching for $\max(\beta)$ via bisection and by searching for $\boldsymbol{q}$ via linear SOS programing techniques.

With the critical similitude ratio (or its estimate) in hand, the maximal set (or its estimate) can be readily calculated. The corresponding critical parameter value $\tilde{\boldsymbol{p}}$ (or its estimate) however, cannot be calculated directly from the solution to the above problems.

5.4 *Failure Domain Approximations*

The algorithm proposed here is similar to the one in Section 4.3 since it iteratively generates the indexed sets $\mathcal{S}_i^{sub}$ (inner approximation to the safe domain), $\mathcal{F}_i^{sub}$ (inner approximation to the failure domain), and Λ_i (region whose containment in $\mathcal{F}$ or $\mathcal{S}$ is to be determined). At any given iteration we first chose a rectangle from Λ_i. By the means presented in Section 5.2 we determine if the quasi-ellipsoid inscribed in this hyper-rectangle is contained in the safe or failure domains. If the ellipsoid is contained in the safe domain, the inner approximation to the safe domain is expanded with this element. If the ellipsoid is contained in the failure domain, the inner approximation to the failure domain is expanded with this element. Otherwise, the rectangle is subdivided into smaller rectangles (see section 2 for two subdivision logics), and these subsets are annexed to Λ_i. The iteration of this procedure, which is stopped when the bounds to the failure probability exceeds a prescribed limit, leads to the desired sequence of approximations. The algorithmic representation of this procedure is as follows.

Start from the very same initial iteration values of the Algorithm in Section 4.3.

1. Let $\mathcal{R}(\bar{\boldsymbol{p}}, \boldsymbol{m})$ be a largest element of Λ_i. Set $\Omega = \mathcal{E}(\bar{\boldsymbol{p}}, \boldsymbol{m}, n)$.

2. Solve Equation (37) if $\sigma = 1$. Otherwise solve Equation (42). Let τ denote either $\tilde{\alpha}$ or $\tilde{\mu}$.

3. If $\tau < 1$, set $\Lambda_{i+1} = (\Lambda_i \setminus \mathcal{R}) \cup \rho(\mathcal{R})$, $\mathcal{S}_{i+1}^{sub} = \mathcal{S}_i^{sub}$, and $\mathcal{F}_{i+1}^{sub} = \mathcal{F}_i^{sub}$. If $\tau \geq 1$ and $\sigma = 1$ let $\Lambda_{i+1} = \Lambda_i \setminus \mathcal{R}$, $\mathcal{S}_{i+1}^{sub} = \mathcal{S}_i^{sub} \cup \Omega$ and $\mathcal{F}_{i+1}^{sub} = \mathcal{F}_i^{sub}$. If $\tau \geq 1$ and $\sigma = -1$ let $\Lambda_{i+1} = \Lambda_i \setminus \mathcal{R}$, $\mathcal{F}_{i+1}^{sub} = \mathcal{F}_i^{sub} \cup \Omega$ and $\mathcal{S}_{i+1}^{sub} = \mathcal{S}_i^{sub}$.

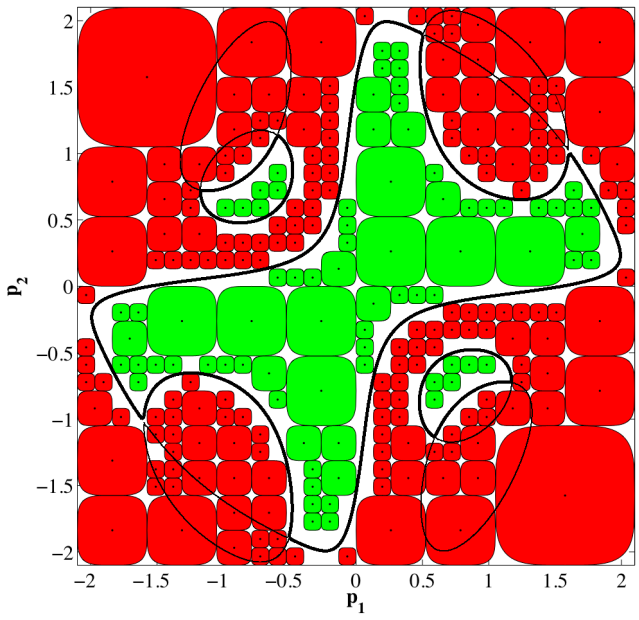

Figure 2: $\mathcal{F}_i^{sub}$ (red), $\mathcal{S}_i^{sub}$ (green), $\partial\mathcal{F}$ (thick line) and individual constraint boundaries (thin lines).

4. Let $\mathcal{F}_{i+1}^{sup} = C(\mathcal{S}_{i+1}^{sub})$. Evaluate $P[\mathcal{F}_{i+1}^{sub}]$ and $P[\mathcal{F}_{i+1}^{sup}]$ or their lower bounds $\psi(\mathcal{F}_{i+1}^{sub})$ and $\psi(\mathcal{F}_{i+1}^{sup})$ depending upon the applicable Theorem (Crespo et al. 2011).

5. If $P[\mathcal{F}_{i+1}^{sub}] \geq 1 - P_{max}$ declare the system acceptable and stop. If $P[\mathcal{F}_{i+1}^{sup}] \leq P_{max}$ declare the system unacceptable and stop. Otherwise, increase i by one, and go to Step (1).

The closing remarks of Section 4.3 apply here as well.

Example 2: The very same problem setup of Example 1 is considered here. Figure 2 shows the subsets comprising $\mathcal{F}^{sub}$ and $\mathcal{S}^{sub}$ for a fixed value of i. Note that the approximations, which result from uniting quartic polynomials ($n = 4$), have voids among neighboring subsets. Further notice that subsets of $\mathcal{F}_i^{sub}$, such as those in the vicinity of $\boldsymbol{p} = [-1, -1.25]^\top$, could have been replaced by a single subset that contains them. Even though this larger subset is contained in $\mathcal{F}$, the containment condition (43-44) is not satisfied. This is the manifestation of $\tilde{\mu} \leq \tilde{\alpha}$. Bounds to the failure probability can be readily calculated from the approximations. Due to the empty space among the neighboring subsets comprising the approximations, these bounds, whose slackness decreases as n increases, are not as tight as those for approximations comprised of hyper-rectangles.

6 CONCLUSIONS

This paper presents an uncertainty analysis framework applicable to systems subject to polynomial requirements. Approaches based on Bernstein expansions and SOS programing are proposed. These and all other methods requiring the exploration of the uncertain parameter space suffer from the curse of dimensionality, hence their computational demands grow exponentially with the number of uncertain parameters. Unfortunately only this space provides the sense of causality required to understand and prevent failure. The high dimensionality of this space along with the inability to guarantee that optimization problems posed there will converge to the global optimum are the main liability of the engineering decisions supported by the outcomes of these methods. This paper proposes techniques that eliminate this liability when the dimension of the uncertain space is moderate. The algorithms proposed allow for data parallelism (i.e., perform computations simultaneously on elements of a subdivision of the master domain). This will help mitigate the formidable challenges of having a large number of uncertain parameters.

REFERENCES

Crespo, L. G., D. P. Giesy, and S. P. Kenny (2008). Robust analysis and robust design of uncertain systems. *AIAA Journal 46*(2), 1–10.

Crespo, L. G., D. P. Giesy, and S. P. Kenny (2009). Reliability-based analysis and design via failure domain bounding. *Structural Safety 31*, 306–315.

Crespo, L. G., S. P. Kenny, and D. P. Giesy (2011, 18-22 September). Uncertainty analysis via failure domain characterization: Unrestricted requirement functions. In *ESREL 2011*, Number 1, Trojes, France.

Jaulin, L., M. Kieffer, O. Didrit, and E. Walter (2001). *Applied Interval Analysis*. London: Springer.

Kall, P. and S. Wallace (1994). *Stochastic Programming*. New York: Wiley.

Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Packard, A., P. Seiler, and G. Balas (2010). Help on SOS [Ask the Experts]. *IEEE Control Systems Magazine 30*(4), 18–23.

Parrillo, P. (2000). *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. Ph.D Dissertation, California Institute of Technology, online at: http://thesis.library.caltach.edu/16471.

Powers, V. and T. Wormann (1998). An algorithm for sum of squares of real polynomials. *Journal of Pure and Applied Linear Algebra 127*, 99–104.

Rackwitz, R. (2001). Reliability analysis, a review and some perspectives. *Structural Safety 23*, 365–395.

Smith, A. P. (2009). Fast construction of constant bound functions for sparse polynomials. *Journal of Global Optimization 43, 2-3*, 445–458.

Zettler, M. and J. Garloff (1998). Robustness analysis of polynomials with polynomial parameter dependency using Bernstein expansion. *IEEE Transactions in Automatic Control 43*(3), 425–431.